

Built-In functions in python.

[Contents](#) [Previous](#) [Next](#)

Goal: Learn about built-in functions in python.

`type`, `str`, `dir` and all the rest of Python's **built-in functions** are grouped into a special module called `__builtin__`. (That's two underscores before and after.) If it helps, you can think of Python automatically executing `from __builtin__ import *` on startup, which imports all the "built-in" functions into the namespace so you can use them directly.

```
# import __builtin__ module to be able to get help on it
import __builtin__

# to quickly check what's inside:
dir(__builtin__)
['ArithError', 'AssertionError', 'AttributeError',
'DeprecationWarning', 'EOFError', 'Ellipsis', 'EnvironmentError',
'Exception', 'False', 'FloatingPointError', 'FutureWarning',
'IOError', 'ImportError', 'IndentationError', 'IndexError',
'KeyError', 'KeyboardInterrupt', 'LookupError', 'MemoryError',
'NameError', 'None', 'NotImplemented', 'NotImplementedError',
'OSSError', 'OverflowError', 'OverflowWarning',
'PendingDeprecationWarning', 'ReferenceError', 'RuntimeError',
'RuntimeWarning', 'StandardError', 'StopIteration',
'SyntaxError', 'SyntaxWarning', 'SystemError', 'SystemExit',
'TabError', 'True', 'TypeError', 'UnboundLocalError',
'UnicodeDecodeError', 'UnicodeEncodeError', 'UnicodeError',
'UnicodeTranslateError', 'UserWarning', 'ValueError', 'Warning',
'ZeroDivisionError', '_', '__debug__', '__doc__', '__import__',
'__name__', 'abs', 'apply', 'basestring', 'bool', 'buffer',
'callable', 'chr', 'classmethod', 'cmp', 'coerce', 'compile',
'complex', 'copyright', 'credits', 'delattr', 'dict', 'dir',
'divmod', 'enumerate', 'eval', 'execfile', 'exit', 'file',
'filter', 'float', 'frozenset', 'getattr', 'globals', 'hasattr',
'hash', 'help', 'hex', 'id', 'input', 'int', 'intern',
'isinstance', 'issubclass', 'iter', 'len', 'license', 'list',
'locals', 'long', 'map', 'max', 'min', 'object', 'oct', 'open',
'ord', 'pow', 'property', 'quit', 'range', 'raw_input', 'reduce',
'reload', 'repr', 'reversed', 'round', 'set', 'setattr',
'slice', 'sorted', 'staticmethod', 'str', 'sum', 'super',
'tuple', 'type', 'unichr', 'unicode', 'vars', 'xrange', 'zip']

# check also the help() function
help(__builtin__)
Help on built-in module __builtin__:

NAME
    __builtin__ - Built-in functions, exceptions, and other objects.

FILE
    (built-in)

DESCRIPTION
    Noteworthy: None is the `nil` object; Ellipsis represents `...` in slices.
```

```

CLASSES
    object
        basestring
            str
            unicode
        buffer
        classmethod
        complex
        dict

# or more specifilcally you can inquire about
# the intrinsic 'file' object
help(__builtin__.file)
Help on class file in module __builtin__:

class file(object)
|   file(name[, mode[, buffering]]) -> file object
|
|   Open a file. The mode can be 'r', 'w' or 'a' for reading (default),
|   writing or appending. The file will be created if it doesn't exist
|   when opened for writing or appending; it will be truncated when
|   opened for writing. Add a 'b' to the mode for binary files.
|   Add a '+' to the mode to allow simultaneous reading and writing.
|   If the buffering argument is given, 0 means unbuffered, 1 means line
|   buffered, and larger numbers specify the buffer size.
|   Add a 'U' to mode to open the file for input with universal newline
|   support. Any line ending in the input file will be seen as a '\n'
|   in Python. Also, a file so opened gains the attribute 'newlines';
|   the value for this attribute is one of None (no newline read yet),
|   '\r', '\n', '\r\n' or a tuple containing all the newline types seen.
|
|   'U' cannot be combined with 'w' or '+' mode.
|
| Note: open() is an alias for file().
|
| Methods defined here:
|
|   __delattr__(...)
|       x.__delattr__('name') <==> del x.name
|
|   __getattribute__(...)
|       x.__getattribute__('name') <==> x.name

```

.....

See also [Python Library Reference](#) for more info on built-in functions.

[Contents](#) [Previous](#) [Next](#)